Preprint typeset in JHEP style - HYPER VERSION

CERN-TH/2001-102
hep-ph/0104145

SOFTSUSY3.1: a program for calculating supersymmetric spectra

B.C. Allanach*,

**DAMTP, CMS, University of Cambridge, Wilberforce road, Cambridge, CB3 0WA, United Kingdom*

Abstract: SOFTSUSY is a program which accurately calculates the spectrum of superparticles in the CP-conserving Minimal Supersymmetric Standard Model (MSSM), with a full flavour mixing structure. The program solves the renormalisation group equations with theoretical constraints on soft supersymmetry breaking terms provided by the user. Weak-scale gauge coupling and fermion mass data (including one-loop finite MSSM corrections) are used as a boundary condition, as well as successful radiative electroweak symmetry breaking. The program can also calculate a measure of fine-tuning. The program structure has been designed to easily generalise to extensions of the MSSM. This article serves as a self-contained guide to prospective users, and indicates the conventions and approximations used.

Keywords: sparticle, MSSM.

Contents

1. Introduction

The Minimal Supersymmetric Standard Model (MSSM) provides an attractive weak-scale extension to the Standard Model. As well as solving the gauge hierarchy problem, it can be motivated by more fundamental models such as various string theories or supersymmetric grand unified theories. The MSSM provides a rich and complicated phenomenology. It predicts many states extra to the Standard Model (sparticles) and their indirect empirical effects and direct detection are vital for verification of the MSSM. Models that are more fundamental than the MSSM can provide stringent constraints upon the way supersymmetry (SUSY) is broken, with important implications for the spectrum which in turn affects the signatures available in experiments. It is therefore desirable to construct a calculational tool which may provide a spectrum and couplings of the MSSM sparticles so that studies of the capabilities of colliders, extraction of high scale parameters (if a signal is observed) and studies of constraints on the models are enabled. We present such a tool (`SOFTSUSY`) in this article.

1.1 The Nature of the Physical Problem

The determination of sparticle masses and couplings of SUSY particles in the R-parity conserving MSSM is the basic problem. Low energy data on Standard Model fermion masses, gauge couplings and electroweak boson masses are to be used as a constraint. SUSY radiative corrections to these inputs from sparticle loops depend upon the sparticle spectrum, and must be calculated. Theoretical constraints on the SUSY breaking parameters from an underlying theory are often imposed at a high renormalisation scale, perhaps resulting from a super gravity or string theory. Often, the theoretical constraints drastically reduce the number of free parameters in the SUSY breaking sector (which numbers over 100 in the unconstrained case). These constraints then make phenomenological analysis tractable by reducing the dimensionality of parameter space sufficiently so that parameter scans over a significant volume of parameter space are possible. Finally, the MSSM parameters must also be consistent with a minimum in the Higgs potential which leads to the observed electroweak boson masses.

This problem has been addressed many times before in the literature (see for example [1, 2, 3, 4, 5]), with varying degrees of accuracy in each part of the calculation. It is our purpose here to provide a tool which will solve the problem with a high accuracy, including state-of-the-art corrections. Similar problems in the context of MSSM extensions[1] have also been studied. In anticipation of new forms of SUSY breaking constraints and new MSSM extensions, we designed the tool to be flexible and easily extended.

1.2 The Program

`SOFTSUSY` has been written in object-oriented C++ but users may use an executable program with input either in the SUSY Les Houches Accord [29] format or from command-line arguments. For users wishing to call `SOFTSUSY` from their own programs, the user interface

[1] By MSSM extension, we mean an extension applicable near the weak scale.

is designed to be C-like to aid users that are unfamiliar with object orientation. Accuracy and ease of generalisation have taken priority over running speed in the design. For example, full three family mass and Yukawa matrices may be employed, rather than the more usual dominant third family approximation. The publicly released codes ISASUGRA (which comprises part of the ISAJET package [5]) and SUSPECT [4] use the dominant third family approximation, for example. The full three-family choice slows the renormalisation group evolution significantly, but will facilitate studies of sparticle or quark mixing. The running time is not foreseen as a bottleneck because it is a matter of a couple of seconds on a modern PC, and will certainly be negligible compared to any Monte-Carlo simulation of sparticle production and decay in colliders. It is possible for the user to specify their own high scale boundary conditions for the soft SUSY breaking parameters without having to change the SOFTSUSY code. For the convenience of most users however, the most commonly used high-scale boundary conditions are included in the package.

The code can be freely obtained from the SOFTSUSY web-page, which currently resides at the address

http://projects.hepforge.org/softsusy/

Installation instructions and more detailed technical documentation of the code may also be found there.

SOFTSUSY is a tool whose output could be used for studies of MSSM sparticle searches [6] by using event generators such as HERWIG [7], or other more theoretical or astrophysical studies. For a review of SUSY tools on offer (which may use the output from SOFTSUSY), see Ref. [8].

1.3 Aims and Layout

The main aims of this article are to provide a manual for the use of SOFTSUSY, to describe the approximations employed and to detail the notation used in order to allow for user generalisation. There have been other articles published on the comparison of the calculation in SOFTSUSY with those of other codes [9], and so we decline from including such information here.

The rest of this paper proceeds as follows: the relevant MSSM parameters are presented in sec. 2. The approximations employed are noted in sec. 3, but brevity requires that they are not explicit. However, a reference is given so that the precise formulae utilised may be obtained in each case. The algorithm of the calculation is also outlined. Technical information related to running and extending the program is placed in appendices. A description of how to run the command-line interface is given in appendix A, including information on the input-file. Sample output from one such run is displayed and explained in appendix B. Appendix C gives a more technical example of a main program, useful if the user wants to call SOFTSUSY from his or her main program. The use of switches and constants is explained in appendix D. Finally, in appendix E, a description of the relevant objects and their relation to each other is presented.

2. MSSM Parameters

In this section, we introduce the MSSM parameters in the `SOFTSUSY` conventions. Translations to the actual variable names used in the source code are shown in appendix E.

2.1 Supersymmetric Parameters

The chiral superfields of the MSSM have the following $G_{SM} = SU(3)_c \times SU(2)_L \times U(1)_Y$ quantum numbers

$$\begin{aligned} &L: (1,2,-\tfrac{1}{2}), \quad \bar{E}: (1,1,1), \qquad Q: (3,2,\frac{1}{6}), \quad \bar{U}: (\bar{3},1,-\frac{2}{3}), \\ &\bar{D}: (\bar{3},1,\tfrac{1}{3}), \quad H_1: (1,2,-\frac{1}{2}), \quad H_2: (1,2,\frac{1}{2}). \end{aligned} \tag{2.1}$$

Then, the superpotential is written as

$$W = \epsilon_{ab}\left[(Y_E)_{ij}L_i^b H_1^a \bar{E}_j + (Y_D)_{ij}Q_i^{bx}H_1^a\bar{D}_{jx} + (Y_U)_{ij}Q_i^{ax}H_2^b\bar{U}_{jx} + \mu H_2^a H_1^b\right] \tag{2.2}$$

Throughout this section, we denote an $SU(3)$ colour index of the fundamental representation by $x, y, z = 1, 2, 3$. The $SU(2)_L$ fundamental representation indices are denoted by $a, b, c = 1, 2$ and the generation indices by $i, j, k = 1, 2, 3$. $\epsilon_{ab} = \epsilon^{ab}$ is the totally antisymmetric tensor, with $\epsilon_{12} = 1$. Note that the sign of μ is identical to the one in `ISASUGRA` [5], but is in the opposite convention to ref. [3]. Presently, real Yukawa couplings only are included. All MSSM running parameters are in the $\overline{DR}$ scheme. The Higgs vacuum expectation values (VEVs) are $\langle H_i^0\rangle = v_i/\sqrt{2}$ and $\tan\beta = v_2/v_1$. g_i are the MSSM $\overline{DR}$ gauge couplings and g_1 is defined in the Grand Unified normalisation $g_1 = \sqrt{5/3}g'$, where g' is the Standard Model hypercharge gauge coupling. Elements of fermion mass matrices are given by

$$(m_u)_{ij} = \frac{1}{\sqrt{2}}(Y_U)_{ij}v_2, \qquad (m_{d,e})_{ij} = \frac{1}{\sqrt{2}}(Y_{D,E})_{ij}v_1 \tag{2.3}$$

for the up quark, down quark and charged lepton matrices respectively.

2.2 SUSY Breaking Parameters

We now tabulate the notation of the soft SUSY breaking parameters. The trilinear scalar interaction potential is

$$V_3 = \epsilon_{ab}\left[\tilde{Q}_{i_L}^{xa}(U_A)_{ij}\tilde{u}_{jx_R}H_2^b + \tilde{Q}_{i_L}^{xb}(D_A)_{ij}\tilde{d}_{jx_R}H_1^a + \tilde{L}_{i_L}^b(E_A)_{ij}\tilde{e}_{j_R}H_1^a + H.c.\right], \tag{2.4}$$

where fields with a tilde are the scalar components of the superfield with the identical capital letter. Also,

$$(A_{U,D,E})_{ij} = (U_A, D_A, E_A)_{ij}/(Y_{U,D,E})_{ij} \tag{2.5}$$

(no summation on i, j) are often referred to in the literature as soft A-parameters.

The scalar bilinear SUSY breaking terms are contained in the potential

$$\begin{aligned} V_2 = {} & m_{H_1}^2 {H_{1a}}^* H_1^a + m_{H_2}^2 {H_{2a}}^* H_2^a + \tilde{Q}_{ixa}^*(m_{\tilde{Q}}^2)_{ij}\tilde{Q}_j^{xa} + \tilde{L}_{ia}^*(m_{\tilde{L}}^2)_{ij}\tilde{L}_j^a + \\ & \tilde{u}_i^x(m_{\tilde{u}}^2)_{ij}\tilde{u}_{jx}^* + \tilde{d}_i^x(m_{\tilde{d}}^2)_{ij}\tilde{d}_{jx}^* + \tilde{e}_i(m_{\tilde{e}}^2)_{ij}\tilde{e}_j^* + \epsilon_{ab}(m_3^2 H_2^a H_1^b + H.c.). \end{aligned} \tag{2.6}$$

For a comparison of these conventions with other popular ones in the literature, see Table 1. In the table, we compare the SOFTSUSY conventions with the SUSY Les Houches Accord [29] and ref. [31] (Martin and Vaughn).

Writing the bino as $\tilde{b}$, $\tilde{w}^{A=1,2,3}$ as the unbroken-SU$(2)_L$ gauginos and $\tilde{g}^{X=1\ldots 8}$ as the gluinos, the gaugino mass terms are contained in the Lagrangian

$$\mathcal{L}_G = \frac{1}{2}\left(M_1\tilde{b}\tilde{b} + M_2\tilde{w}^A\tilde{w}^A + M_3\tilde{g}^X\tilde{g}^X\right) + h.c. \tag{2.7}$$

SOFTSUSY	SLHA	Martin and Vaughn
$Y^{U,D,E}$	$Y^{U,D,E}$	$(Y^{U,D,E})^T$
U_A, D_A, E_A	$T_{U,D,E}$	$h^T_{U,D,E}$
$m^2_{\tilde{Q},\tilde{L}}$	$m^2_{\tilde{Q},\tilde{L}}$	$m^2_{\tilde{Q},\tilde{L}}$
$m^2_{\tilde{u},\tilde{d},\tilde{e}}$	$m^{2\ T}_{\tilde{u},\tilde{d},\tilde{e}}$	$m^2_{\tilde{u},\tilde{d},\tilde{e}}$
μ	μ	μ
m_3^2	$B\mu$	m_3^2

Table 1: Comparison of conventions between SOFTSUSY and the literature.

2.3 Tree-Level Masses

Here we suppress any gauge indices and follow the notation of ref. [3] closely. The Lagrangian contains the neutralino mass matrix as $-\frac{1}{2}\tilde{\psi}^{0T}\mathcal{M}_{\tilde{\psi}^0}\tilde{\psi}^0$ + h.c., where $\tilde{\psi}^0 = (-i\tilde{b}, -i\tilde{w}^3, \tilde{h}_1, \tilde{h}_2)^T$ and

$$\mathcal{M}_{\tilde{\psi}^0} = \begin{pmatrix} M_1 & 0 & -M_Z c_\beta s_W & M_Z s_\beta s_W \\ 0 & M_2 & M_Z c_\beta c_W & -M_Z s_\beta c_W \\ -M_Z c_\beta s_W & M_Z c_\beta c_W & 0 & -\mu \\ M_Z s_\beta s_W & -M_Z s_\beta c_W & -\mu & 0 \end{pmatrix}. \tag{2.8}$$

We use s and c for sine and cosine, so that $s_\beta \equiv \sin\beta$, $c_\beta \equiv \cos\beta$ and $s_W(c_W)$ is the sine (cosine) of the weak mixing angle. The 4 by 4 neutralino mixing matrix is an orthogonal matrix O with real entries, such that $O^T\mathcal{M}_{\tilde{\psi}^0}O$ is diagonal. The neutralinos χ_i^0 are defined such that their absolute masses increase with increasing i. Some of their mass values can be negative.

We make the identification $\tilde{w}^\pm = (\tilde{w}^1 \mp i\tilde{w}^2)/\sqrt{2}$ for the charged winos and $\tilde{h}_1^-, \tilde{h}_2^+$ for the charged higgsinos. The Lagrangian contains the chargino mass matrix as $-\tilde{\psi}^{-T}\mathcal{M}_{\tilde{\psi}^+}\tilde{\psi}^+ +$ $h.c.$, where $\tilde{\psi}^+ = (-i\tilde{w}^+,\ \tilde{h}_2^+)^T$, $\tilde{\psi}^- = (-i\tilde{w}^-,\ \tilde{h}_1^-)^T$ and

$$\mathcal{M}_{\tilde{\psi}^+} = \begin{pmatrix} M_2 & \sqrt{2}\,M_W s_\beta \\ \sqrt{2}\,M_W c_\beta & \mu \end{pmatrix}. \tag{2.9}$$

This matrix is then diagonalised by 2 dimensional rotations through angles θ_L, θ_R in the following manner:

$$\begin{pmatrix} c_{\theta_L} & s_{\theta_L} \\ -s_{\theta_L} & c_{\theta_L} \end{pmatrix}\mathcal{M}_{\tilde{\psi}^+}\begin{pmatrix} c_{\theta_R} & -s_{\theta_R} \\ s_{\theta_R} & c_{\theta_R} \end{pmatrix} = \begin{pmatrix} m^+_{\chi_1} & 0 \\ 0 & m^+_{\chi_2} \end{pmatrix} \tag{2.10}$$

where $m^+_{\chi_i}$ could be negative, with the mass parameter of the lightest chargino being in the top left hand corner.

At tree level the gluino mass, $m_{\tilde{g}}$, is given by M_3.

Strong upper bounds upon the intergenerational scalar mixing exist [10] and in the following we assume that such mixings are negligible. The tree-level squark and slepton

mass squared values for the family i are found by diagonalising the following mass matrices $\mathcal{M}^2_{\tilde{f}}$ defined in the $(\tilde{f}_{iL}, \tilde{f}_{iR})^T$ basis:

$$\begin{pmatrix} (m^2_{\tilde{Q}})_{ii} + m^2_{ui} + (\frac{1}{2} - \frac{2}{3}s^2_W)M^2_Z c_{2\beta} & m_{u_i}\,((A_U)_{ii} - \mu\cot\beta) \\ m_{u_i}\,((A_U)_{ii} - \mu\cot\beta) & (m^2_{\tilde{u}})_{ii} + m^2_{u_i} + \frac{2}{3}s^2_W M^2_Z c_{2\beta} \end{pmatrix}, \tag{2.11}$$

$$\begin{pmatrix} (m^2_{\tilde{Q}})_{ii} + m^2_{d_i} - (\frac{1}{2} - \frac{1}{3}s^2_W)M^2_Z c_{2\beta} & m_{d_i}\,((A_D)_{ii} - \mu\tan\beta) \\ m_{d_i}\,((A_D)_{ii} - \mu\tan\beta) & (m^2_{\tilde{d}})_{ii} + m^2_{d_i} - \frac{1}{3}s^2_W M^2_Z c_{2\beta} \end{pmatrix}, \tag{2.12}$$

$$\begin{pmatrix} (m^2_{\tilde{L}})_{ii} + m^2_{e_i} - (\frac{1}{2} - s^2_W)M^2_Z c_{2\beta} & m_{e_i}\,((A_E)_{ii} - \mu\tan\beta) \\ m_{e_i}\,((A_E)_{ii} - \mu\tan\beta) & (m^2_{\tilde{e}})_{ii} + m^2_{e_i} - s^2_W M^2_Z c_{2\beta} \end{pmatrix}, \tag{2.13}$$

m_f, e_f are the mass and electric charge of fermion f respectively. The mixing of the first two families is suppressed by a small fermion mass, which we approximate to zero. The sfermion mass eigenstates are given by

$$\begin{pmatrix} m_{\tilde{f}_1} & 0 \\ 0 & m_{\tilde{f}_2} \end{pmatrix} = \begin{pmatrix} c_f & s_f \\ -s_f & c_f \end{pmatrix} \mathcal{M}^2_{\tilde{f}} \begin{pmatrix} c_f & -s_f \\ s_f & c_f \end{pmatrix} \tag{2.14}$$

where c_f is the cosine of the sfermion mixing angle, $\cos\theta_f$, and s_f the sine. θ_f are set in the convention that the two mass eigenstates are in no particular order and $\theta_f \in [-\pi/4, \pi/4]$. The sneutrinos of one family are not mixed and their masses are given by

$$m^2_{\tilde{\nu}_i} = (m^2_{\tilde{L}})_{ii} + \frac{1}{2}M^2_Z c_{2\beta}. \tag{2.15}$$

The CP-even gauge eigenstates (H^0_1, H^0_2) are rotated by the angle α into the mass eigenstates $(H^0\, h^0)$ as follows,

$$\begin{pmatrix} H^0 \\ h^0 \end{pmatrix} = \begin{pmatrix} c_\alpha & s_\alpha \\ -s_\alpha & c_\alpha \end{pmatrix} \begin{pmatrix} \frac{1}{\sqrt{2}}\Re H^0_1 \\ \frac{1}{\sqrt{2}}\Re H^0_2 \end{pmatrix}. \tag{2.16}$$

$m_{h^0} < m_{H^0}$ by definition, and $\alpha \in [-\pi/4, 3\pi/4]$. The CP-odd and charged Higgs masses are

$$m^2_{A^0} = m^2_3(\tan\beta + \cot\beta), \qquad m^2_{H^\pm} = m^2_{A^0} + M^2_W \tag{2.17}$$

at tree level.

3. Calculation

We now show the algorithm used to perform the calculation. Standard Model parameters (fermion and gauge boson masses, the fine structure constant $\alpha(M_Z)$, the Fermi constant from muon decay G^μ_F and $\alpha_3(M_Z)$) are used as constraints. The soft SUSY breaking parameters and the superpotential parameter μ are then the free parameters. However, in what follows, $|\mu|$ is constrained by M_Z and $\tan\beta$ is traded for m_3 as an input parameter. Therefore, the total list of unconstrained input parameters is: any fundamental soft SUSY breaking breaking parameters (except m^2_3), $\tan\beta$ and the sign of μ. First we describe the evolution of the low-energy Standard Model input parameters below M_Z, then detail the rest of the algorithm.

3.1 Below M_Z

$\alpha(M_Z)$, $\alpha_s(M_Z)$ are first evolved to 1 GeV using 3 loop QCD and 1 loop QED [11, 12, 13] with step-function decoupling of fermions at their running masses. We have checked that the contribution from 2-loop matching [14] is negligible; the effect of 3-loop terms in the renormalisation group equations is an order of magnitude larger. Then, the two gauge couplings and all Standard Model fermion masses except the top mass are run to M_Z. The β functions of fermion masses are taken to be zero at renormalisation scales below their running masses. The parameters at M_Z are used as the low energy boundary condition in the rest of the evolution.

3.2 Initial Estimate

The algorithm proceeds via the iterative method, and therefore an approximate initial guess of MSSM parameters is required. For this, the third family $\overline{DR}$ Yukawa couplings are approximated by

$$h_t(Q) = \frac{m_t(Q)\sqrt{2}}{v \sin\beta}, \qquad h_{b,\tau}(Q) = \frac{m_{b,\tau}(Q)\sqrt{2}}{v \cos\beta}, \tag{3.1}$$

where $v = 246.22$ GeV is the Standard Model Higgs VEV and $Q = m_t(m_t)$ is the renormalisation scale. The $\overline{MS}$ values of fermion masses are used for this initial estimate. The fermion masses and α_s at the top mass are obtained by evolving the previously obtained fermion masses and gauge couplings from M_Z to m_t (with the same accuracy). The electroweak gauge couplings are estimated by $\alpha_1(M_Z) = 5\alpha(M_Z)/(3c_W^2)$, $\alpha_2(M_Z) = \alpha(M_Z)/s_W^2$. Here, s_W is taken to be the on-shell value. These two gauge couplings are then evolved to m_t with 1-loop Standard Model β functions, including the effect of a light higgs (without decoupling it). In this initial guess, no SUSY threshold effects are calculated. The gauge and Yukawa couplings are then evolved to the unification scale M_X with the one-loop MSSM β functions, where the user-supplied boundary condition on the soft terms is applied. Also, $\mu(M_X) = \mathrm{sgn}(\mu) \times 1$ GeV and $m_3(M_X) = 0$ are imposed. These initial values are irrelevant; they are overwritten on the next iteration by more realistic boundary conditions. $\mu(M_X)$ is set to be of the correct sign because its sign does not change through renormalisation.

The whole system of MSSM soft parameters and SUSY couplings is then evolved to 1-loop order to M_Z. At M_Z, the tree-level electroweak symmetry breaking (EWSB) conditions are applied [6] to predict μ and m_3. The masses and mixings of MSSM super particles are then calculated at tree-level order by using the SUSY parameters (and m_3) calculated at M_Z. The resulting set of MSSM parameters is then used as the initial guess for the iterative procedure described below.

3.3 Gauge and Yukawa Couplings

Figure 1 shows the iterative procedure, starting from the the top. The whole calculation is currently performed in the real full three family approximation, i.e. all Yukawa couplings are set to be real, but quark mixing is incorporated. First of all, the one-loop radiative

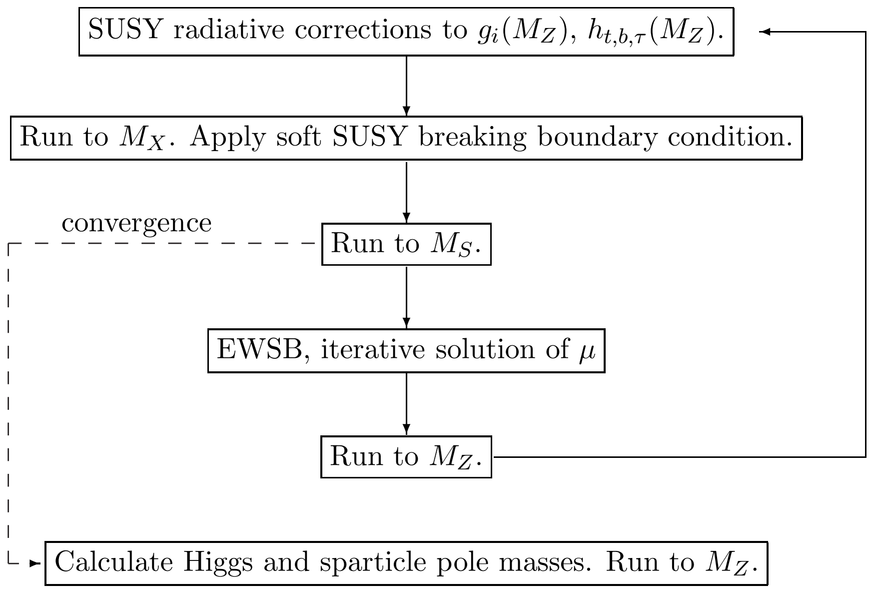

Figure 1: Iterative algorithm used to calculate the SUSY spectrum. Each step (represented by a box) is detailed in the text. The initial step is the uppermost one. M_S is the scale at which the EWSB conditions are imposed, as discussed in the text. M_X is the scale at which the high energy SUSY breaking boundary conditions are imposed.

corrections are applied to the gauge and third-family Yukawa couplings. For these, we rely heavily on ref. [3] by Bagger, Matchev, Pierce and Zhang (BMPZ)[2]. In the threshold corrections, we use running $\overline{DR}$ masses and parameters at the relevant scale, unless denoted otherwise in the text. $m_t(M_Z)$ is calculated with 2-loop QCD [15] and the full one-loop supersymmetric contributions to $m_t(M_Z)$, including logarithmic and finite contributions (eqs. (D.16)-(D.18) of BMPZ):

$$\begin{aligned} m_t(M_Z)^{\overline{DR}}_{MSSM} = \Sigma_t^{BMPZ} + m_t^{pole} \Bigg(1 - \frac{\alpha_s(M_Z)}{3\pi}(5-3L) + \\ \alpha_s^2(M_Z)(-0.538 + \frac{43}{24\pi^2} L - \frac{3}{8\pi^2} L^2) \Bigg), \end{aligned} \tag{3.2}$$

where $L \equiv \ln(m_t^2(M_Z)^{\overline{DR}}_{MSSM}/M_Z^2)$. We denote the BPMZ corrections *without* the one-loop QCD part as Σ_t^{BMPZ}. These corrections are necessary because the region of valid EWSB is very sensitive to m_t [6]. To calculate $m_b(M_Z)^{\overline{DR}}$, we first calculate the Standard Model $\overline{DR}$ value from the $\overline{MS}$ one [15, 32]

$$m_b(M_Z)^{\overline{DR}}_{SM} = m_b(M_Z)^{\overline{MS}}_{SM} \left(1 - \frac{\alpha_s^{\overline{DR}}}{3\pi} - \frac{23 \alpha_s^{\overline{DR}^2}}{72\pi^2} + \frac{3g^2}{128\pi^2} + \frac{13 g_1^2}{1152\pi^2} \right). \tag{3.3}$$

We then add the leading one-loop supersymmetric corrections

$$m_b(M_Z)^{\overline{DR}}_{MSSM} = m_b(M_Z)^{\overline{DR}}_{SM} / (1 + \Delta^b_{SUSY}). \tag{3.4}$$

[2]Whenever a reference to an equation in BMPZ is made, it is understood that the sign of μ must be reversed.

The contributions to Δ^b_{SUSY} are included in full from eq. D.18 of BMPZ (neglecting the term proportional to e, since that is already included in the QED calculation of the SM $m_b(M_Z)$). Both finite and leading logarithmic corrections are included. After the Standard Model $\overline{MS}$ bar of m_τ is converted to the $\overline{DR}$ value via

$$m_\tau(M_Z)^{\overline{DR}}_{SM} = m_\tau(M_Z)^{\overline{MS}}_{SM} \left(1 - \frac{3}{128\pi^2}(g_1^2 - g_2^2)\right). \tag{3.5}$$

The full one-loop MSSM corrections from the appendix of BPMZ (aside from the photon contribution, since that has already been included in $m_\tau(M_Z)^{\overline{MS}}_{SM}$) are then used to correct $m_\tau(M_Z)^{\overline{DR}}_{SM}$:

$$m_\tau(M_Z)^{\overline{DR}}_{MSSM} = m_\tau(M_Z)^{\overline{DR}}_{SM}(1 + \Sigma_\tau). \tag{3.6}$$

The one-loop $\overline{DR}$ values for $m_t(M_Z)$, $m_b(M_Z)$, $m_\tau(M_Z)$ are then substituted with the $\overline{DR}$ value of $v(Q)$ into eq. (2.3) to calculate the third family $\overline{DR}$ Yukawa couplings at M_Z. $v(Q)$ is run to two-loops. The other diagonal elements of the Yukawa matrices are set by eq. (2.3) but with fermion masses replaced by the $\overline{MS}$ values.

The default option is to perform the calculation in the dominant third-family approximation, where all elements of Yukawa matrices expect for the (3,3) elements are set to zero. There are also options described in appendix D for performing the calculation in the unmixed 3-family approximation or the fully-mixed 3-family case. If a flavour-mixing option is chosen (see section E.9), the Yukawa couplings are then mixed using the "standard parameterisation" of the CKM matrix [17] with CP-violating phase set either to zero or π, whichever results in a positive entry for $(V_{CKM})_{13}$ (also known as V_{ub}):

$$V_{CKM} = \begin{pmatrix} c_{12}c_{13} & s_{12}c_{13} & ps_{13} \\ -s_{12}c_{23} - pc_{12}s_{23}s_{13} & c_{12}c_{23} - ps_{12}s_{23}s_{13} & s_{23}c_{13} \\ s_{12}s_{23} - pc_{12}c_{23}s_{13} & -c_{12}s_{23} - ps_{12}c_{23}s_{13} & c_{23}c_{13} \end{pmatrix}, \tag{3.7}$$

where $s_{ij} \equiv \sin\theta_{ij}$, $c_{ij} \equiv \cos\theta_{ij}$ and $p = \pm 1$. Sign conventions are automatically chosen such that diagonal entries and entries above the diagonal in V_{CKM} are positive. Note that V_{CKM} is a member of $O(3)$, i.e. $V^{-1}_{CKM} = V^T_{CKM}$. While complex phase effects are obviously not taken into account in eq. 3.7, it is hoped that the magnitudes of the main quark mixing effects will be. In fact, using the central values for θ_{ij} given by the particle data group [17], the magnitudes of all elements in V_{CKM} in the first row and last column are exactly reproduced. Of the other entries, $|V_{cs}|$ is accurate at the 2×10^{-5} level and $|V_{cd}|$ at the 4×10^{-4} level: surely negligible for most practical purposes. $|V_{ts}|$ is also quite accurate (to 1.2%), but it should be noted that $|V_{td}|$ is wrong by around 50%. Any flavour physics effects sensitive to $|V_{td}|$ is therefore subject to this large uncertainty on its value issuing from `SOFTSUSY`[3]. The up (by default), or down Yukawa couplings at M_Z are mixed in the weak eigenbasis via

$$(Y_U)' = V^T_{CKM}(Y^U)V_{CKM}, \qquad (Y_D)' = V_{CKM}(Y^D)V^T_{CKM} \tag{3.8}$$

[3]It is hoped in the future to include complex phases in Yukawa matrices, sfermion soft mass squared terms and trilinear scalar couplings.

where the primed Yukawa matrix is in the weak eigenbasis and the unprimed is in the mass eigenbasis.

Full one-loop corrections to $g_i(M_Z)$ are included. The treatment of electroweak gauge couplings follows from appendix C of BMPZ, and includes: two-loop corrections from the top, electroweak boson and the lightest CP-even Higgs. The pole value of m_t is used in the calculation of the W and Z self energy in order to calculate $\sin\theta_w$, since this is what is assumed in the two-loop corrections. We use the fine structure constant $\alpha(M_Z)^{\overline{MS}}$, the Z-boson mass M_Z and the Fermi decay constant G_μ as inputs. M_W is predicted from these inputs. Because the EWSB constraints tend to depend sensitively upon $g_{1,2}(M_Z)$, accurate values for them are determined iteratively. An estimate of the $\overline{DR}$ value of s^2_W is used to yield a better estimate until the required accuracy is reached (usually within 3 or 4 iterations). The QCD coupling is input as $\alpha_s(M_Z)^{\overline{MS}}$ and is modified by gluino, squark and top loops as in eqs. (2),(3) of BMPZ in order to obtain the MSSM $\overline{DR}$ value.

3.4 MSSM Renormalisation

All soft breaking and SUSY parameters are then evolved to the scale

$$M_S \equiv \sqrt{m_{\tilde{t}_1}(M_S) m_{\tilde{t}_2}(M_S)}, \tag{3.9}$$

where [18] the scale dependence of the electroweak breaking conditions is smallest. Throughout the iteration described here, the renormalisation group evolution (RGE) employs three family, 2-loop MSSM β functions for the supersymmetric parameters [2]. $\tan\beta$ and the Higgs VEV parameter v are also run to two-loop order in the Feynman gauge, although the Higgs VEV RGE is missing terms $\mathcal{O}(g_2^4, g_2^2 g_1^2, g_1^4)/(16\pi^2)^2$ [19, 20]. The program can be made much faster by switching certain two-loop terms off (scalar sfermion masses and trilinear couplings) as described in section C. There is no step-function decoupling of sparticles: this is taken into account at leading logarithmic order in the radiative corrections previously calculated at M_Z and in the calculation of the physical sparticle spectrum at M_S, described below. All β functions are real and include 3 family (and mixing) contributions. If no flavour mixing is present in the model specified by the user, the 2-loop parts of the RGEs switch to the dominant 3rd-family version, where the lighter two families Yukawa couplings are neglected.

3.5 Electroweak Symmetry Breaking

The Higgs VEV parameter $v(M_S)$ is set by:

$$v^2(M_S) = 4 \frac{M_Z^2 + \Re \Pi^T_{ZZ}(M_S)}{g_2^2(M_S) + 3 g_1^2(M_S)/5}, \tag{3.10}$$

where M_Z is the pole Z mass and Π^T_{ZZ} is the transverse Z self-energy. The full one-loop EWSB conditions at this scale are then employed to[4] calculate $m_3(M_S)$ and $\mu(M_S)$. $\mu(M_S)$

[4]Note that there is also an option to extract $m_{H_1}(M_{SUSY})$ and $m_{H_2}(M_{SUSY})$ from input $\mu(M_{SUSY})$ and m_A(pole) values, see section E.8.

requires an iterative solution because the tadpoles depend upon the value of μ assumed. The symmetry breaking condition for μ can be phrased as [3]

$$\mu^2 = \frac{1}{2}\left(\tan 2\beta \left[m^2_{\bar{H}_2}\tan\beta - m^2_{\bar{H}_1}\cot\beta\right] - M^2_{\bar{Z}}\right), \tag{3.11}$$

where $m^2_{\bar{H}_i} = m^2_{H_i} - t_i/v_i$, $M^2_{\bar{Z}} = M^2_Z + \Re\Pi^T_{ZZ}(M^2_Z)$ is the running Z mass and t_i are the tadpole contributions. The value of μ coming from the tree-level EWSB condition (eq. 3.11, with $\Re\Pi^T_{ZZ} = t_i = 0$) is utilised as an initial guess, then the one-loop contributions in the tadpoles and self-energy terms are added to provide a new value of $\mu(M_S)$. Two-loop terms $\mathcal{O}(\alpha_t^2)$, $\mathcal{O}(\alpha_b\alpha_\tau)$, $\mathcal{O}(\alpha_b^2)$, $\mathcal{O}(\alpha_b\alpha_s)$, $\mathcal{O}(\alpha_t\alpha_s)$, $\mathcal{O}(\alpha_\tau^2)$ and $\mathcal{O}(\alpha_t\alpha_b)$ are also included in the tadpoles [21, 26]. The tadpole corrections are then calculated using the new value of $\mu(M_S)$ and the procedure is repeated until it converges to a given accuracy. $m_3(M_S)$ is then determined by input the value of $\mu(M_S)$ into the EWSB condition

$$m_3^2 = \frac{s_{2\beta}}{2}\left(m^2_{\bar{H}_1} + m^2_{\bar{H}_2} + 2\mu^2\right). \tag{3.12}$$

The ensemble of MSSM parameters are then evolved using the β functions described above to the user supplied scale M_X. If gauge-unification has been specified as a boundary condition, the current estimate of M_X is revised to leading log order to provide a more accurate value upon the next iteration:

$$M_X^{new} = M_X e^{\frac{g_2(M_X)-g_1(M_X)}{g_1'(M_X)-g_2'(M_X)}}, \tag{3.13}$$

where primes denote derivatives calculated to 2-loop order. The user-supplied boundary conditions are then imposed upon the soft terms before the model is evolved back down to M_S. The super particle mass spectrum is determined at this scale. Because μ and m_3 are more scale independent at M_S as opposed to some other scale, the Higgs, neutralino and chargino masses also ought to be more scale independent by determining them at this scale.

3.6 MSSM Spectrum

In the following description of the approximations involved in the calculation of the super particle spectrum, it is implicit that where masses appear, their $\overline{DR}$ values are employed. The running value of $s_W(\mu) = e(\mu)/g_2(\mu)$ is also employed. In loop corrections to sparticle masses, the Yukawa couplings of the first two families are set to zero, being highly suppressed compared to those of the third family. All sparticle masses are calculated with the full SUSY one-loop BPMZ corrections at the scale M_{SUSY}. Most sparticle masses are calculated at external momenta equal to their $\overline{DR}$ mass $m(M_{SUSY})$.

The physical gluino mass is calculated to full one-loop order as follows. The running parameters are evaluated at renormalisation scale $\mu = M_{SUSY}$ and external momentum $p = M_3(\mu)$ in the following corrections:

$$\Delta_{\tilde{g}}(\mu) = \frac{g_3(\mu)^2}{16\pi^2}\left(15 + 9\ln\left(\frac{\mu^2}{p^2}\right) - \sum_q \sum_{i=1}^{2} B_1(p, m_q, m_{\tilde{q}_i}, \mu) - \right.$$

$$\left.\sum_{q=t,b} \frac{m_q}{M_3(\mu)} s_{2\theta_q} \left[B_0(p, m_q, m_{\tilde{q}_1}, \mu) - B_0(p, m_q, m_{\tilde{q}_2}, \mu) \right] \right) . \quad (3.14)$$

The Passarino-Veltman functions $B_{0,1}$ are given in appendix B of BMPZ. The physical gluino mass is then given by

$$m_{\tilde{g}} = M_3(M_{SUSY}) \left(1 + \Delta_{\tilde{g}}(M_{SUSY})\right) . \quad (3.15)$$

The gluino mass is allowed to be negative, as is the case in mAMSB, for example. Of course the kinematic mass is just the absolute value, and the phase may be rotated away, altering the phases of some of the Feynman rules. Negative masses for neutralinos can also be rotated away in this way.

For mixed sparticles (sbottoms, staus, neutralinos and charginos), the external momentum is equal to the $\overline{DR}$ value of the diagonalised mass. All mixing angles and matrices are defined such that they diagonalise the one-loop corrected mass matrix evaluated at the *minimum* $\overline{DR}$ mass eigenvalue. When flavour mixing is used, at tree-level, the full flavour structure is present in sfermion masses. Loop corrections are added only to the equal family pieces, not to intra-family mixing pieces as an approximation. All of the `SOFTSUSY` loop corrections also neglect intra-family mixing themselves.

The pseudo-scalar Higgs mass m_{A^0} and CP-even Higgs masses m_{h^0}, m_{H^0} are determined to full one-loop order as in eq. (E.6) of BMPZ in order to reduce their scale dependence, which can be large [22]. The zero-momentum $\mathcal{O}(\alpha_t^2)$, $\mathcal{O}(\alpha_b \alpha_\tau)$, $\mathcal{O}(\alpha_b^2)$, $\mathcal{O}(\alpha_b \alpha_s)$, $\mathcal{O}(\alpha_t \alpha_s)$, $\mathcal{O}(\alpha_\tau^2)$, $\mathcal{O}(\alpha_t \alpha_b)$ 2-loop corrections are also included in $m_{A^0}, m_{h^0}, m_{H^0}$ [23, 24, 25, 26]. All one-loop corrections are included in the determination of the charged Higgs pole mass. Every Higgs mass is determined at an external momentum scale equal to its $\overline{DR}$ mass.

Finally, the running MSSM parameters are evolved back down to M_Z. The whole process is iterated as shown in figure 1, until the $\overline{DR}$ sparticle masses evaluated at M_S all converge to better than the desired fractional accuracy (`TOLERANCE`), which may be set by the user in the main program or input file.

3.7 Fine Tuning

We now detail the fine-tuning calculation. As lower bounds on super partner masses are pushed up by colliders, m_{H_1} and m_{H_2} may be forced to be much larger than M_Z if they are related to the other super particle masses, as is the case for example in the case of minimal super gravity. If we re-phrase eq. (3.11) as

$$M_{\bar{Z}}^2 = -2\mu^2 + \tan 2\beta \left[m_{\bar{H}_2}^2 \tan\beta - m_{\bar{H}_1}^2 \cot\beta \right] , \quad (3.16)$$

we see that the terms on the right-hand side must have some degree of cancellation in order to reproduce the observed value of M_Z. But μ has a different origin to the SUSY breaking parameters and the balancing appears unnatural. Various measures have been proposed in order to quantify the apparent cancellation, for example ref.s [27, 28]. The definition of

naturalness c_a of a 'fundamental' parameter a employed here is [28]

$$c_a \equiv \left| \frac{\partial \ln M_Z^2}{\partial \ln a} \right|. \tag{3.17}$$

From a choice of a set of fundamental parameters defined at the scale M_X: $\{a_i\}$, the fine-tuning of a particular model is defined to be $c = \max(c_a)$. $\{a_i\}$ are any parameters in the user supplied boundary condition on the soft supersymmetry breaking parameters augmented by $h_t(M_X), \mu(M_X)$ and $m_3(M_X)$. The derivatives in eq. (3.17) are calculated by numerically finding the derivative of $M_Z^{pole} = \hat{M}_Z + \Re\Pi^T_{ZZ}(M_Z^2)$ in eq. (3.11). The input parameters are changed slightly (one by one), then the MSSM parameter ensemble is run from M_X to M_S where the sparticle mass spectrum is determined along with the corresponding $\overline{MS}$ Higgs VEV parameter $v^2 \equiv v_1^2 + v_2^2$. First of all, $\tan\beta(M_S)$ is determined by inverting eq. (3.12) and the resulting value is utilised in a version of eq. (3.11) inverted to give M_Z^{pole} in terms of the other parameters. The resulting value of M_Z^{pole} is the prediction for the new changed input parameters, and its derivative is determined by examining its behaviour as the initial changes in input parameters tend to zero.

A. Running SOFTSUSY

A main program is supplied which produces an executable called `softpoint.x`, is included in the `SOFTSUSY` distribution. For the calculation of the spectrum of single points in parameter space, we recommend the SLHA [29] input/output option. The user must modify a file (e.g. `lesHouchesInput`, as provided in the standard distribution) that specifies the input parameters. The user may then run the code with

```
./softpoint.x leshouches < lesHouchesInput
```

In this case, the output will also be in SLHA format. Such output can be used for input into other programs which subscribe to the accord, such as `SDECAY` [34], `PYTHIA` [35] (for simulating sparticle production and decays at colliders) or `mirOMEGAs` [36] (for calculating the relic density of neutralinos, $b \to s\gamma$ and $\mu \to e\gamma$), for example. For further details on the necessary format of the input file, see ref. [29] and appendix A.1.

If the user desires to quickly run a single parameter point in AMSB, mSUGRA or GMSB parameter space, but does not wish to use the SLHA, the following options are available:

```
./softpoint.x sugra <m0> <m12> <a0> <tanb> <mgut> <sgnMu>
./softpoint.x amsb <m0> <m32> <tanb> <mgut> <sgnMu>
./softpoint.x gmsb <n5> <mMess> <lambda> <tanb> <sgnMu>
```

Bracketed entries should be replaced by the desired numerical values, (in GeV if they are dimensionful). The program will provide output from one point in mSUGRA, mAMSB or GMSB. If `<mgut>` is specified as `unified` on input, `SOFTSUSY` will determine `mgut` to be the scale that $g_1(M_{GUT}) = g_2(M_{GUT})$, usually of order 10^{16} GeV. If `<mgut>` is set to be `msusy`, the SUSY breaking parameters will be set at M_{SUSY}.

For users that are not familiar with `C++`, we note that the executable interface allows the calculation at just one parameter point in SUSY breaking space. If scans are required,

the user can either call SOFTSUSY from a shell script or use a system call from a main C program to the executable. Alternatively, a main program showing an example of a scan is provided. C++ beginners should note in the following that "method" means function, that objects contain a list of data structures and functions and that for a user to access (change or reference) the data encoded in an object, one of its functions should be called. Such functions are given in tables of the following appendices.

A.1 Input file

If, as recommended, the SLHA option is used for input, the user may add a SOFTSUSY-specific block to the input file in the following format, with bracketed entries replaced by double precision values:

```
Block SOFTSUSY     # SOFTSUSY specific inputs
  1   <TOLERANCE>  # desired fractional accuracy in output
  2   <MIXING>     # quark mixing option
  3   <PRINTOUT>   # gives additional verbose output during calculation
  4   <QEWSB>      # change electroweak symmetry breaking scale
  5   <INCLUDE_2_LOOP_SCALAR_CORRECTIONS> # Full 2-loop running in RGEs
```

The fractional numerical precision on masses and couplings output by SOFTSUSY is better than TOLERANCE, which sets the accuracy of the whole calculation. The iteration of each physical SUSY particle mass is required to converge to a fractional accuracy smaller than TOLERANCE. Sub-iterations are required to converge to a better accuracy than $10^{-2} \times$TOLERANCE for s_W and $10^{-4} \times$TOLERANCE for μ. The accuracy of the Runge-Kutta RGE changes from iteration to iteration but is proportional to the value of TOLERANCE. Values between 10^{-2} and 10^{-6} are common, lower values mean that SOFTSUSY takes significantly longer to perform the calculation.

The next parameter MIXING determines what M_Z boundary condition will be used for the quark Yukawa matrix parameters. MIXING=-1.0 sets all Yukawa couplings to zero at M_Z except for the third-family ones (dominant third-family approximation). MIXING=0.0 sets the quark mixings to zero but includes the first two family's diagonal terms. MIXING=1.0,2.0 sets all the mixing at M_Z to be in the up-quark or down-quark sector respectively, as in eq. (3.8).

Setting PRINTOUT to a non-zero value gives additional information on each successive iteration. If PRINTOUT>0, a warning flag is produced when the overall iteration finishes. The predicted values of M_Z^{pole} and $\tan\beta(M_S)$ after iteration convergence are also output[5]. The level of convergence, $\mu(M_S)$, $m_3^2(M_S)$ and M_Z are output with each iteration, as well as a flag if the object becomes non-perturbative. PRINTOUT>1 produces output on the fine-tuning calculation. The predicted values of M_Z^{pole} and $\tan\beta(M_S)$ are output with each variation in the initial inputs. A warning flag is produced when a negative-mass squared scalar is present. PRINTOUT>2 prints output on the sub-iterations that determine $\mu(M_S)$ and $s_W(M_S)$, and flags the nature of any tachyons encountered. Values PRINTOUT>0 are only required if additional diagnostics are required for debugging purposes.

[5]Note that the input value of $\tan\beta$ is the value at M_Z.

QEWSB may be used to multiplicatively change the scale M_S at which the Higgs potential is minimised and sparticle masses calculated, or it may alternatively be used as a fixed scale. This can be useful if one wants to examine the scale dependence of the results [30]. Setting QEWSB=$x < M_Z/1$ GeV, sets $M_S = x\sqrt{m_{\tilde{t}_1}(M_S) m_{\tilde{t}_2}(M_S)}$. Values from 0.5 to 2 are common. If $M_S < M_Z$ results from the above expression, $M_S = M_Z$ is used. If on the other hand a *fixed scale* is desired from M_S, setting QEWSB$> M_Z/1$ GeV in the input results in $M_S =$QEWSB GeV being fixed.

If INCLUDE_2_LOOP_SCALAR_CORRECTIONS is switched off (0) as in the default case, 2-loop RGEs [31] are used for the Higgs and gaugino masses, μ, Yukawa and gauge couplings but 1-loop RGEs are used for other MSSM parameters. Switching on the 2-loop corrections (1) results in a full 2-loop RGE evolution, but slows the calculation by a factor of approximately three.

If softpoint.x is used *without* the SLHA interface, default Standard Model inputs are used from the files def.h and lowe.h. The low energy data is encoded in a QedQcd object and must be provided. The default numbers supplied and contained in the QedQcd object are are given in units of GeV and running masses are in the $\overline{MS}$ scheme. For the bottom and top masses, *either* the running mass *or* the pole mass must be supplied as an input. The type of mass not given for input is calculated by SOFTSUSY at the 3-loop QCD level. We recommend, along with ref. [32], that the running mass be used for m_b since there are smaller theoretical errors in the extraction of this quantity from experiment than the pole mass. The scale dependent quantities in this object are then evolved to M_Z by the method toMz, to provide the low-scale empirical boundary condition for the rest of the calculation. The MSSM spectrum calculated depends most crucially upon $M_Z, \alpha(M_Z), \alpha_s(M_Z)$ and input third family fermion masses.

B. Sample Output

For the recommended SLHA option, the conventions for the output are fully explained in Ref. [29]. We therefore present the non-SLHA compliant SOFTSUSY output for a single example calculation which can be run by the command

```
./softpoint.x sugra 100 250 -100 10 unified 1
```

The parameters here have been chosen from SUGRA point SPS1a [33]. The output obtained was

```
SOFTSUSY SUGRA calculation
mgut = 2.393247e+16
----------------------------------------------------------
Gravitino mass M3/2: 0.000000e+00
Msusy: 4.667356e+02 MW: 8.039821e+01
Data set:
mU: 1.390102e-03  mC: 6.353475e-01  mt: 1.638989e+02  mt^pole: 1.714000e+02
mD: 2.764078e-03  mS: 6.051876e-02  mB: 2.884361e+00  mb(mb): 4.200000e+00
mE: 5.026670e-04  mM: 1.039356e-01  mT: 1.751633e+00  mb^pole: 4.938977e+00
aE: 1.279250e+02  aS: 1.176000e-01   Q: 9.118760e+01  mT^pole: 1.776990e+00
```

```
loops: 3        thresholds: 1
----------------------------------------------------------
SUSY breaking MSSM parameters at Q: 9.118760e+01
 UA(3,3):
-5.932554e-03  0.000000e+00  0.000000e+00
 0.000000e+00 -2.711465e+00  0.000000e+00
 0.000000e+00  0.000000e+00 -4.956895e+02
 UD(3,3):
-1.491378e-01  0.000000e+00  0.000000e+00
 0.000000e+00 -3.265318e+00  0.000000e+00
 0.000000e+00  0.000000e+00 -1.296751e+02
 UE(3,3):
-7.385460e-03  0.000000e+00  0.000000e+00
 0.000000e+00 -1.527041e+00  0.000000e+00
 0.000000e+00  0.000000e+00 -2.624597e+01
 mQLsq(3,3):
 3.494430e+05  0.000000e+00  0.000000e+00
 0.000000e+00  3.494406e+05  0.000000e+00
 0.000000e+00  0.000000e+00  2.900203e+05
 mURsq(3,3):
 3.298251e+05  0.000000e+00  0.000000e+00
 0.000000e+00  3.298228e+05  0.000000e+00
 0.000000e+00  0.000000e+00  2.126351e+05
 mDRsq(3,3):
 3.276634e+05  0.000000e+00  0.000000e+00
 0.000000e+00  3.276608e+05  0.000000e+00
 0.000000e+00  0.000000e+00  3.232042e+05
 mLLsq(3,3):
 3.902935e+04  0.000000e+00  0.000000e+00
 0.000000e+00  3.902814e+04  0.000000e+00
 0.000000e+00  0.000000e+00  3.866974e+04
 mSEsq(3,3):
 1.857551e+04  0.000000e+00  0.000000e+00
 0.000000e+00  1.857307e+04  0.000000e+00
 0.000000e+00  0.000000e+00  1.784658e+04
m3sq: 2.434581e+04 mH1sq: 3.148341e+04 mH2sq: -1.555013e+05
Gaugino masses(1,3):
 9.815767e+01  1.885414e+02  6.307201e+02
Supersymmetric parameters at Q: 9.118760e+01
 Y^U(3,3):
 7.921595e-06  0.000000e+00  0.000000e+00
 0.000000e+00  3.620574e-03  0.000000e+00
 0.000000e+00  0.000000e+00  9.027082e-01
 Y^D(3,3):
 1.575130e-04  0.000000e+00  0.000000e+00
 0.000000e+00  3.448705e-03  0.000000e+00
 0.000000e+00  0.000000e+00  1.479975e-01
 Y^E(3,3):
```

```
 2.864485e-05  0.000000e+00  0.000000e+00
 0.000000e+00  5.922845e-03  0.000000e+00
 0.000000e+00  0.000000e+00  1.025530e-01
higgs VEV: 2.494075e+02 tan beta: 1.000000e+01 smu: 3.464099e+02
g1: 4.592134e-01 g2: 6.430272e-01 g3: 1.139490e+00
thresholds: 3 #loops: 2
---------------------------------------------------------
Physical MSSM parameters:
mh^0: 1.095488e+02 mA^0: 3.926544e+02 mH^0: 3.936189e+02 mH^+-: 4.011008e+02
alpha: -1.142202e-01
sneutrinos(1,3):
 1.861463e+02  1.861434e+02  1.852040e+02
mU~(2,3):
 5.634127e+02  5.634110e+02  5.858263e+02
 5.465897e+02  5.465880e+02  3.999008e+02
mD~(2,3):
 5.688527e+02  5.688511e+02  5.140942e+02
 5.464146e+02  5.464128e+02  5.462770e+02
mE~(2,3):
 2.023270e+02  2.023417e+02  2.062159e+02
 1.440522e+02  1.440445e+02  1.345834e+02
thetat: -5.929027e-01 thetab: 3.216005e-01 thetatau: -2.855378e-01
mGluino:  6.066386e+02
charginos(1,2):
 1.795193e+02  3.762019e+02
thetaL: -4.203557e-01 thetaR: -2.429011e-01
neutralinos(1,4):
 9.691276e+01  1.797635e+02 -3.562353e+02  3.743742e+02
neutralino mixing matrix (4,4):
 9.848209e-01  1.091056e-01 -6.158751e-02 -1.201277e-01
-5.833843e-02  9.386643e-01  9.171542e-02  3.272526e-01
 1.532705e-01 -2.827469e-01  6.943969e-01  6.437199e-01
-5.685500e-02  1.644979e-01  7.110613e-01 -6.812487e-01
lsp is neutralino of mass 9.691276e+01 GeV
---------------------------------------------------------
```

Firstly, the output details the input parameters, starting with `MIXING` and `TOLERANCE`. After the output of the input `QedQcd` object and the value of M_{GUT}, the result of the iteration algorithm in sec. 3 is output in the form of a `MssmSoftsusy` object, unless the user is using the SLHA option (in which case, the output will be in SLHA format). The soft SUSY breaking parameters were defined in sec. 2.2, and are listed in appendix E.7. First of all, the soft SUSY breaking parameters are displayed. In order, they are the up, down and charged lepton trilinear scalar matrices (in units of GeV). Next come the mass squared values of the left-handed squarks, right-handed up squarks, right-handed down squarks, left-handed sleptons, right-handed charged sleptons in GeV^2. m_3^2, $m_{H_1}^2$, $m_{H_2}^2$ and gaugino mass parameters follow. The parameter, $m_{3/2}$ (not used here) is the VEV of a compensator superfield in anomaly-mediation and completes the SUSY breaking parameter list.

Supersymmetric parameters (see sections 2.1,E.5) are displayed next: Yukawa matrices Y^U,Y^D,Y^E, $\tan\beta$,g_i, the accuracy level of the calculation, bilinear superpotential μ parameter, renormalisation scale and maximum number of loops used for RGE.

Physical MSSM parameters follow. The pole masses and mixing parameters are previously listed in sec. 2.3, and are detailed in appendix E.7. All masses are in units of GeV, and all mixing angles are given in radians. Respectively, there is: m_{h^0}, m_{A^0}, m_{H^0}, m_{H^0} and α. Scalar sparticle masses $m_{\tilde{\nu}}$, $m_{\tilde{u}}$, $m_{\tilde{d}}$, $m_{\tilde{e}}$ follow, as well as the mixing angles θ_t, θ_b, θ_τ. The gauginos are listed (in order): $m_{\tilde{g}}$, $m_{\chi^\pm}$, θ_L, θ_R, m_{χ^0} and O. The $\overline{DR}$ Higgs VEV $v(M_S)$ is then listed, followed by the $\overline{MS}$ low energy data used as a boundary condition at M_Z. Finally, the identity of the lightest supersymmetric particle is shown, together with its mass.

Problems (if any) with the parameter point are printed in the final line (none in this case, indicating a viable point).

B.1 Problem flags

Any associated problems such as negative mass-squared scalars or inconsistent EWSB are flagged at the end of the output. None of these are printed because the SPS1a point displayed has none of these problems. We now list the problems, indicating their meaning:

- If `No convergence` appears, then `SOFTSUSY` is indicating that it didn't achieve the accuracy of `TOLERANCE` within less than 40 iterations. The output of the code is therefore to be considered unreliable and it is not clear from the output whether the point is allowed or disallowed, despite the presence or absence of other warning messages. This error flag often appears near the boundary of electroweak symmetry breaking, (where $\mu(M_{SUSY}) = 0$)), where the iterative algorithm is not stable. To calculate the position of the electroweak symmetry boundary, one should interpolate between regions a small distance away from it.

- `Non-perturbative` indicates that `SOFTSUSY` encountered couplings reaching Landau poles when evolving, and could not calculate any further. Any results obtained using perturbation theory (for example those of `SOFTSUSY`) therefore cannot be trusted.

- `Infra-red quasi fixed point breached` indicates that the parameter point is at a Landau pole of a Yukawa coupling. This should not be a problem provided no other errors are flagged.

- `muSqWrongSign` indicates that the Higgs minimisation conditions imply that $\mu^2 < 0$, meaning that the desired electroweak minimum is not present in the model. The model is ruled out.

- The `tachyon` flag occurs when a scalar particle other than the Higgs has acquired a negative mass squared, when $M_Z^2 < 0$ or when a pole Higgs masses is imaginary. The model is ruled out.

- `noRhoConvergence` is flagged when `SOFTSUSY` cannot calculate the ρ parameter and determine gauge couplings from data, typically because of tachyons or infinities that

have crept into the calculation. The other problems are serious enough to rule the model out.

- `higgsUfb` and `bProblem` indicate that the desired electroweak minimum is in fact a saddle point of the potential, thus the model is ruled out.

- `mgutOutOfBounds` is flagged if the value of the gauge unification scale predicted by eq. 3.13 is outside the range 10^4 GeV $< M_X < 5 \times 10^{17}$ GeV. The GUT-scale has been set to the appropriate limit, and the `SOFTSUSY` numbers cannot be trusted.

Thus flags other than `No convergence`, `Infra-red quasi fixed point breached` or `Non-perturbative` indicate an unphysical minimum of the scalar potential, effectively ruling the model point out.

C. Sample Program

We now present the sample program from which it is possible to run `SOFTSUSY` in a simple fashion. The program we presents here performs a scan in the variable $\tan\beta$, with other parameters as in mSUGRA point SPS1a. It then prints the four pole Higgs masses as a function of $\tan\beta$ in the standard output channel. If there are any problems with the parameter point, the program prints out these instead of the Higgs masses. The most important features of the objects are described in appendix E. The sample program `main.cpp` has the following form:

```
#include <iostream>
#include "mycomplex.h"
#include "def.h"
#include "linalg.h"
#include "lowe.h"
#include "rge.h"
#include "softsusy.h"
#include "softpars.h"
#include "susy.h"
#include "utils.h"
#include "numerics.h"

/// global variable declaration
namespace softsusy {
  /// no quark mixing and no verbose output
  int MIXING = 0, PRINTOUT = 0;
  /// fractional accuracy required
  double TOLERANCE = 1.0e-3;
  /// decay constant of muon
  double GMU = 1.16637e-5;
  /// there are two possible conventions: if QEWSB > MZ, its value is assumed
  /// in GeV and used as a constant MSUSY. Otherwise, it MULTIPLIES the usual
  /// MSUSY value, of root(mstop1 mstop2)
  double QEWSB = 1.0;
```

```
  /// Do we include 2-loop RGEs of *all* scalar masses and A-terms, or only the
  /// scalar mass Higgs parameters? (Other quantities all 2-loop anyway): the
  /// default in SOFTSUSY 3 is to include all 2-loop terms
  bool INCLUDE_2_LOOP_SCALAR_CORRECTIONS = true;
  /// number of loops used to calculate Higgs mass and tadpoles. They should be
  /// identical for a consistent calculation
  int numHiggsMassLoops = 2, numRewsbLoops = 2;
  /// global pole mass of MZ in GeV - MZCENT is defined in def.h
  double MZ = MZCENT;
  /// end of global variable declaration
}

int main() {
 /// Sets format of output: 6 decimal places
  outputCharacteristics(6);

  cerr << "SOFTSUSY" << SOFTSUSY_VERSION
       << " test program, Ben Allanach 2002\n";
  cerr << "If you use SOFTSUSY, please refer to B.C. Allanach,\n";
  cerr << "Comput. Phys. Commun. 143 (2002) 305, hep-ph/0104145\n";
  cerr << "B.C. Allanach and M. Bernhardt, arxiv:0903.1805 [hep-ph]\n\n";

  /// Parameters used: mSUGRA parameters
  double m12 = 250., a0 = -100., mGutGuess = 2.0e16, tanb = 10.0, m0 = 100.;
  int sgnMu = 1;      ///< sign of mu parameter
  int numPoints = 10; ///< number of scan points

  QedQcd oneset;      ///< See "lowe.h" for default definitions parameters

  /// most important Standard Model inputs: you may change these and recompile
  double alphasMZ = 0.1187, mtop = 173.4, mbmb = 4.2;
  oneset.setAlpha(ALPHAS, alphasMZ);
  oneset.setPoleMt(mtop);
  oneset.setMass(mBottom, mbmb);

  oneset.toMz();      ///< Runs SM fermion masses to MZ

  /// Print out the SM data being used, as well as quark mixing assumption and
  /// the numerical accuracy of the solution
  cout << "# Low energy data in SOFTSUSY: MIXING=" << MIXING << " TOLERANCE="
       << TOLERANCE << endl << oneset << endl;

  /// Print out header line
  cout << "# tan beta   mh           mA           mH0          mH+-\n";

  int i;
  /// Set limits of tan beta scan
  double startTanb = 3.0, endTanb = 50.0;
```

```
  /// Cycle through different points in the scan
  for (i = 0; i<=numPoints; i++) {

    tanb = (endTanb - startTanb) / double(numPoints) * double(i) +
      startTanb; // set tan beta ready for the scan.

    /// Preparation for calculation: set up object and input parameters
    MssmSoftsusy r;
    DoubleVector pars(3);
    pars(1) = m0; pars(2) = m12; pars(3) = a0;
    bool uni = true; // MGUT defined by g1(MGUT)=g2(MGUT)

    /// Calculate the spectrum
    r.lowOrg(sugraBcs, mGutGuess, pars, sgnMu, tanb, oneset, uni);

    /// check the point in question is problem free: if so print the output
    if (!r.displayProblem().test())
      cout << tanb << " " << r.displayPhys().mh0 << " "
           << r.displayPhys().mA0 << " "
           << r.displayPhys().mH0 << " "
           << r.displayPhys().mHpm << endl;
    else
      /// print out what the problem(s) is(are)
      cout << tanb << " " << r.displayProblem() << endl;
  }
}
```

After an initial introductory print-out, the variables specifying the supersymmetry breaking parameters are specified. For these, the same notation as appendix A is used. Next, the important Standard Model inputs are defined and combined with the defaults already present in the `QedQcd` object. The top running mass is calculated from the pole mass and Standard Model fermion masses and gauge couplings are then run up to M_Z with the method `toMz`.

name	arguments
`sugraBcs`	m_0, $m_{1/2}$, A_0
`amsbBcs`	$m_{3/2}$, m_0
`gmsbBcs`	n_5, m_{mess}, Λ

Table 2: SUSY breaking boundary conditions available to the user, detailing arguments in order. The asterisk denotes additional information in the text.

If `gaugeUnification=true`, `softsusy` will determine `mGutGuess` from electroweak gauge unification, using the `mGutGuess` value supplied as an initial guess. The user can supply a void function that sets the supersymmetry breaking parameters from an input `DoubleVector`. In the sample code given above, this function is `sugraBcs` and is applied to the `MssmSoftsusy` object at the scale M_{GUT}, which will be determined by the user (but `mGutGuess` will be used as an initial guess). Other examples of available boundary conditions are given in Table 2. The user must supply a `DoubleVector` containing the numerical values of the arguments, correctly ordered as in Table 2. `sugraBcs`, for example, calls the `MssmSoftsusy` method `standardSugra(m0, m12, a0)`, which sets all scalar masses equal to `m0`, all gaugino masses to `m12` and all trilinear

scalar couplings to `a0`, in the standard universal fashion. If the user desires to write his or her own boundary condition, it must conform to the prototype
`void userDefinedBcs(MssmSoftsusy & m, const DoubleVector & inputs)`
The method `lowOrg` drives the calculation after, which $\tan\beta$ and the Higgs masses are printed out.

D. Switches and Constants

The file `def.h` contains the switches and constants. If they are changed, the code must be recompiled in order to use the new values. Table 3 shows the most important parameters in `def.h`, detailing the default values that the constants have. G_μ and M_Z have been obtained using the latest particle data group numbers [17].

variable	default	description
`ARRAY_BOUNDS_CHECKING`	off	Vector and Matrix bounds checking
`EPSTOL`	10^{-11}	Underflow accuracy
`GMU`	$1.16637\ 10^{-5}$	G_μ, Fermi constant from muon decay
`MZCENT`	91.1876	Pole mass of the Z^0 boson M_Z.

Table 3: Switches and constants. Starred entries have more explanation in the text. G_μ is in units of GeV^2 and M_Z in GeV.

E. Object Structure

We now go on to sketch the objects and their relationship. This is necessary information for generalisation beyond the MSSM. Only methods and data which are deemed important for prospective users are mentioned here, but there are many others within the code itself.

E.1 Linear Algebra

The `SOFTSUSY` program comes with its own linear algebra classes: `Complex`, `DoubleVector`, `DoubleMatrix`, `ComplexVector`, `ComplexMatrix`. Constructors of the latter four objects involve the dimensions of the object, which start at 1. `Complex` objects are constructed with their real and imaginary parts respectively. For example, to define a vector $a_{i=1,2,3}$, a matrix $m_{i=1\dots3,j=1\dots4}$ of type `double` and a Complex number $b = 1 - i$:

```
DoubleVector a(3);
DoubleMatrix m(3, 4);
Complex      b(1.0, -1.0);
```

Obvious algebraic operators between these classes (such as multiplication, addition, subtraction) are defined with overloaded operators `*`, `+`, `-` respectively. Elements of the vector and matrix classes are referred to with brackets `()`. `DoubleVector` and `DoubleMatrix` classes are contained within each of the higher level objects that we now describe.

E.2 General Structure

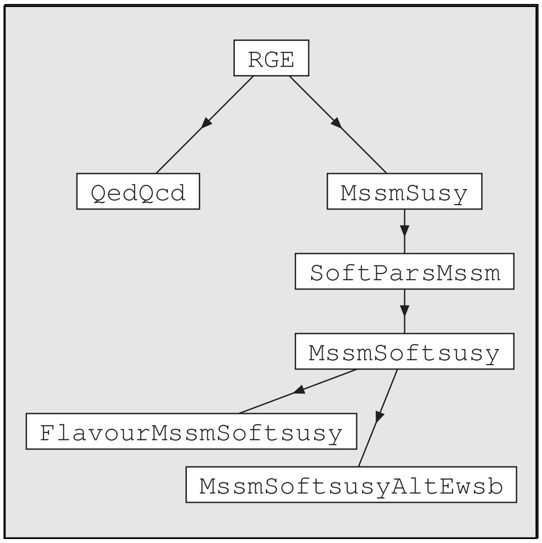

Figure 2: Heuristic high-level object structure of `SOFTSUSY`. Inheritance is displayed by the lines.

From a RGE point of view, a particular quantum field theory model consists of a set of couplings and masses defined at some renormalisation scale μ. A set of β functions describes the evolution of the parameters and masses to a different scale μ'. This concept is embodied in an *abstract* `RGE` object, which contains the methods required to run objects of derived classes to different renormalisation scales. The other objects displayed in figure 2 are particular instances of `RGE`, and therefore inherit from it. `QedQcd` objects consist of data on the quark and lepton masses and gauge couplings. They contain the β functions for running in an effective QED$\times$QCD theory below m_t. An object of class `MssmSusy` contains the Yukawa couplings, and the three gauge couplings of the MSSM. It also contains the superpotential μ term (not to be confused with the renormalisation scale), $\tan\beta$, the ratio of the two Higgs doublet VEVs as well as $v = \sqrt{v_1^2 + v_2^2}$. Its β functions are valid in the exact SUSY limit of the MSSM. The major part of the code resides within the `MssmSoftsusy` class. Objects of this type have all the functionality of `MssmSusy`, with soft SUSY breaking terms and theoretical boundary conditions contained in the inherited class `SoftParsMssm`. It also contains an object of type `QedQcd` which contains weak scale empirical data. Code in the `MssmSoftsusy` class organises and performs the main part of the calculation. `MssmSoftsusyAltEwsb` objects are a slight variant of `MssmSoftsusy`: it takes non-universal Higgs mass boundary conditions at the SUSY breaking scale. `FlavourMssmSoftsusy` objects are also slight variants: they have full flavour-mixed output and input. In the following, we provide basic information on classes so that users may program using `SOFTSUSY`. Highly detailed and technical documentation on the program may be obtained from the `SOFTSUSY` website.

E.3 `RGE` Class

The data and important methods in `RGE` are presented in table 4. Each of the higher level objects described in this appendix have explicitly named `display` and `set` methods that are used to access or change the data contained within each object. In table 4 (as in the following tables in this section), these accessing methods are listed on the same row as the relevant data variable.

The `RGE` method `runto(mup, eps)` will automatically run any derived object to the scale `mup` with a fractional accuracy of evolution `eps`. In order to define this evolution, any object that inherits from an `RGE` must contain three methods: `display`, `set`, `beta` shown in table 4. `DoubleVector display() const` must return a vector containing all masses and couplings of the object, in some arbitrary user-defined order. `void set(const DoubleVector & v)`

must set these couplings given a `DoubleVector v` defined in the same order as the display function. `DoubleVector beta() const` must then return the β functions in a `DoubleVector` defined as

$$\beta_i = \frac{da_i}{d \ln \mu}, \tag{E.1}$$

where a_i denotes any mass or coupling of the model. The ordering of the a_i must be identical in each of the three methods.

data variable		methods
`double mu=` μ	renormalisation scale (GeV)	`setMu` `displayMu`
`int numpars`	number of scale dependent parameters	`setPars` `howMany`
`int loops`	accuracy of RGE	`setLoops` `displayLoops`
`int thresholds`	accuracy level of threshold computation	`setThresholds` `displayThresholds`

method	function
`DoubleVector display()`	displays all running parameters (*)
`void set(DoubleVector)`	sets all running parameters (*)
`DoubleVector beta`	displays beta functions of all running parameters (*)
`runto`	runs object to new value of `mu`

Table 4: Abstract `RGE` class. (*) indicates that derived objects *must* contain these methods (see text).

E.4 `QedQcd` Class

The `QedQcd` class contains a `DoubleVector` of quark and lepton $\overline{MS}$ masses ($m_f = m_{u,d,e,c,s,\mu,t,b,\tau}(\mu)$), as shown in table 5. Its contents may be printed to standard output or read from standard input (with the same format in each case) by using the operators `<<` or `>>`, as can all the non-abstract objects mentioned in this section. The methods `toMz()`, `toMt()` act on an initial object defined with each fermion mass m_f defined at a scale

$$Q' = \max(1 \text{ GeV}, m_f(m_f)) \tag{E.2}$$

and gauge couplings at M_Z.

E.5 `MssmSusy` Class

The operators `<<`, `>>` have been overloaded to write or read a `MssmSusy` object to/from a file stream. Table 6 shows the data variables and important methods contained in the class. For the Yukawa and gauge couplings, methods exist to either set (or display) one element or a whole matrix or vector of them.

data variable		methods
`DoubleVector a` $\alpha(\mu), \alpha_s(\mu)$	$\overline{MS}$ gauge couplings	`setAlpha` `displayAlpha`
`DoubleVector m` $m_f(\mu)$	running fermion masses vector $(1\ldots 9)$ (GeV)	`setMass` `displayMass`
`double mtPole, mbPole` `double mtauPole` $m_t^{pole}, m_b^{pole}, m_\tau^{pole}$	pole top/bottom/tau mass (GeV)	`setPoleMt`, `setPoleMb` `setPoleMtau` `displayPoleMt` `displayPoleMb` `displayPoleMtau`

method	function
`runGauge`	runs gauge couplings *only*
`toMt, toMZ`	runs fermion masses and gauge couplings from Q' to m_t^{pole} or M_Z

Table 5: `QedQcd` class. Q' is defined in the text.

E.6 `SoftParsMssm` Class

The operators `<<`, `>>` have been overloaded to write or read a `softParsMssm` object to/from a file stream. Table 7 shows the data variables and important methods contained in the class. `addAmsb()` adds anomaly mediated supersymmetry breaking terms [37] to the model's soft parameters. Such terms are proportional to the VEV of a compensator superfield, so $m_{3/2}$ in table 7 must have been set before `addAmsb` is used. `minimalGmsb(int n5, double lambda, double mMess)` applies the messenger scale `mMess` boundary conditions to the soft masses in minimal gauge-mediated supersymmetry breaking [38]. `n5` denotes the number of $5 \oplus \bar{5}$ messenger fields that are present and `lambda`(Λ) is as described in ref. [38].

E.7 `MssmSoftsusy` Class

`MssmSoftSusy` objects contain a structure `sPhysical` encapsulating the physical information on the superparticles (pole masses and physical mixings), as shown in table 8. Another structure of type `drBarPars` inherits from `sPhysical` but instead contains information on $\overline{DR}$ masses and mixing angles. `MssmSoftSusy` objects also contain one of these structures for calculational convenience: the information is used in order to calculate loop corrections to various masses. As table 9 shows, a method `mpzCharginos` returns the 2 by 2 complex diagonalisation matrices U, V that result in positive $\overline{DR}$ chargino masses, as defined in ref. [3]. The method `mpzNeutralinos` is present in order to convert O to the complex matrix N defined in ref. [3] that would produce only positive $\overline{DR}$ neutralino masses. This information, as well as $\overline{DR}$ third family fermion masses are stored in the `drBarPars` structure. Another structure within `MssmSoftsusy` of type `sProblem` flags various potential problems with the object, for example the lack of radiative EWSB or negative mass squared scalars (excluding the Higgs mass squared parameters). This structure is shown in table 10. In addition, the method `test` prints out if any of the possible data variables flagging problems

data variable		methods
`DoubleMatrix u, d, e` $(Y_U)_{ij}, (Y_D)_{ij}, (Y_E)_{ij}$	Yukawa couplings (3 by 3 matrix)	`setYukawaElement` `setYukawaMatrix` `displayYukawaElement` `displayYukawaMatrix`
`DoubleVector g` g_i	MSSM gauge couplings $(1 \ldots 3)$ vector	`setAllGauge` `setGaugeCoupling` `displayGauge` `displayGaugeCoupling`
`double smu` μ	bilinear Higgs superpotential parameter	`setSusyMu` `displaySusyMu`
`double tanb` $\tan\beta$	ratio of Higgs VEVs (at current renormalisation scale)	`setTanb` `displayTanb`
`double hVev` v	Higgs VEV	`setHvev` `displayHvev`

method	function
`setDiagYukawas`	calculates and sets all diagonal Yukawa couplings given fermion masses and a Higgs VEV
`getMasses`	calculates quark and lepton masses from Yukawa couplings
`getQuarkMixing`	mixes quark Yukawa couplings from mass to weak basis
`getQuarkMixedYukawas`	sets all entries of quark Yukawa couplings given fermion masses, Higgs VEV and CKM matrix

Table 6: `MssmSusy` class.

are true. The `higgsUfb` flag is true if

$$m_{H_1}^2 + 2\mu^2 + m_{H_2}^2 - 2|m_3^2| < 0 \tag{E.3}$$

is not satisfied, implying that the desired electroweak minimum is either a maximum or a saddle-point of the tree-level Higgs potential [2]. The contents of `sPhysical` and `sProblem` can be output with overloaded `<<` operators. `noConvergence` means that the desired accuracy was not reached. `nonPerturbative` or `irqfp` flags the existence of a Landau pole in the renormalisation group evolution, and the calculation is not perturbatively reliable so any results should be discarded. All other problems except `noConvergence` and `nonPerturbative` should be considered as grounds for ruling the model out. `noRhoConvergence` occurs when the pseudo-scalar Higgs A^0 has a negative mass squared (i.e. an invalid electroweak vacuum).

`MssmSoftsusy` data variables and accessors can be viewed in table 11 and the most important high-level methods are displayed in table 12. The operators `<<`, `>>` have been overloaded to write or read `MssmSoftsusy` objects or `sPhysical` structures to/from a file stream. The driver routine for the RGE evolution and unification calculation is

data		methods
`double m32` $m_{3/2}$	compensator VEV* (GeV)	`setM32` `displayGravitino`
`DoubleVector mGaugino` $M_{1,2,3}$	$(1 \ldots 3)$ vector of gaugino mass parameters	`setGauginoMass` `displayGaugino`
`DoubleMatrix ua,da,ea` U_A, D_A, E_A	(3 by 3) matrix of trilinear soft terms (GeV)	`setTrilinearElement` `displayTrilinearElement` `displaySoftA`
`DoubleMatrix mQLsq` `mURsq,mDRsq,mLLsq` `mSEsq` $(m^2_{\tilde{Q}_L})$, $(m^2_{\tilde{u}_R})$, $(m^2_{\tilde{d}_R})$, $(m^2_{\tilde{L}_L})$, $(m^2_{\tilde{e}_R})$	(3 by 3) matrices of soft SUSY breaking masses (GeV^2)	`setSoftMassElement` `setSoftMassMatrix` `displaySoftMassSquared`
`double m3sq,mH1sq,mH2sq` m_3^2, $m^2_{H_1}$, $m^2_{H_2}$	Bilinear Higgs parameters (GeV, GeV^2, GeV^2)	`setM3Squared` `setMh1Squared` `setMh2Squared` `displayM3Squared` `displayMh1Squared` `displayMh2Squared`

method	function
`standardSugra`	Sets all universal soft terms
`universalScalars`	Sets universal scalar masses
`universalGauginos`	Sets universal gaugino masses
`universalTrilinears`	Sets universal soft breaking trilinear couplings
`addAmsb`	Adds AMSB soft terms to current object*
`minimalGmsb`	Gauge-mediated soft terms used as boundary conditions*

Table 7: `SoftParsMssm` class data and methods.

```
double MssmSoftsusy::lowOrg
(void (*boundaryCondition)(MssmSoftsusy &, const DoubleVector &),
 double mxGuess, const DoubleVector & pars, int sgnMu, double tanb, const QedQcd &
 oneset, bool gaugeUnification)
```

The user-supplied `boundaryCondition` function sets the soft parameters according to the elements of the supplied `DoubleVector` at `mxGuess`, as discussed in appendix C. If `gaugeUnification` is `true`, the scale that unifies the electroweak gauge couplings is used and returned by the function. If `gaugeUnification` is `false`, the function simply returns `mxGuess`. `pars` contains a `DoubleVector` of soft SUSY breaking parameters to be applied as the theoretical boundary condition. `sgnMu` is the sign of the superpotential μ parameter, `tanb` is the value of $\tan\beta(M_Z)$ required and `oneset` contains the M_Z scale low energy data.

The fine tuning (as defined in sec. 3) can be calculated with the method

```
DoubleVector MssmSoftsusy::fineTune(void (*boundaryCondition)
```

data variable	description
`double mh0,mA0,mH0,mHpm`	$h^0, A^0, H^0, H^\pm$masses
`DoubleVector msnu`	vector of $m_{\tilde{\nu}_{i=1\ldots3}}$ masses
`DoubleVector mch,mneut`	vectors of $m_{\chi^\pm{}_{i=1\ldots2}}$, $m_{\chi^0{}_{i=1\ldots4}}$ respectively
`double mGluino`	gluino mass $m_{\tilde{g}}$
`DoubleMatrix mixNeut`	4 by 4 orthogonal neutralino mixing matrix O
`double thetaL, thetaR`	$\theta_{L,R}$ chargino mixing angles
`double thetat, thetab`	$\theta_{t,b}$ sparticle mixing angles
`double thetatau, thetaH`	θ_τ, α sparticle and Higgs mixing angles
`DoubleMatrix mu, md, me`	(2 by 3) matrices of up squark, down squark and charged slepton masses
`double t1OV1Ms, t2OV2Ms`	tadpoles t_1/v_1 and t_2/v_2 evaluated at M_S

Table 8: `sPhysical` structure. Masses are pole masses, and stored in units of GeV. Mixing angles are in radian units.

data variable	description
`double mt, mb, mtau` $m_t(Q), m_b(Q), m_\tau(Q)$	Third family fermion masses
`DoubleVector mnBpmz, mchBpmz` $m_{\chi_i^0}, m_{\chi_i^\pm}$	Absolute neutralino and chargino masses (1...4, 1...2) vectors
`ComplexMatrix nBpmz` N	Neutralino mixing matrix (4 by 4 complex matrix)
`ComplexMatrix uBpmz, vBpmz` U, V	Chargino mixing matrices (2 by 2 complex matrices)

name	function
`mpzNeutralino`	Gives mixing matrices required to make neutralino masses positive*
`mpzChargino`	Gives mixing matrices required to make chargino masses positive*

Table 9: `drBarPars` structure. Masses are in the $\overline{DR}$ scheme, and stored in units of GeV. Mixing angles are in radian units. Functions marked with an asterisk are mentioned in the text.

```
(MssmSoftsusy &, const DoubleVector &), const DoubleVector
 & bcPars, double mx) const
```

This function should only be applied to an `MssmSoftsusy` object which has been processed by `lowOrg`. `mx` is the unification scale and `boundaryCondition` is the function that sets the unification scale soft parameters, as discussed above. In derived objects, the virtual method `methodBoundaryCondition` may be used to set data additional to `MssmSoftsusy` from the `boundaryCondition` function. The method outputs the fine-tuning of a parameter $a_{i=1\ldots n}$

data variable	flags
`irqfp`	in a region with a Landau pole
`noConvergence`	the main iteration routine doesn't converge
`noRhoConvergence`	the ρ iterative routine doesn't converge
`tachyon`	a non-Higgs scalar has negative mass squared
`muSqWrongSign`	μ^2 from eq. (3.11) negative
`b`	B from eq. (3.12) has incorrect sign
`higgsUfb`	eq. (E.3) is not satisfied
`nonperturbative`	a Landau pole was reached below the unification scale

Table 10: `sProblem` structure. All data variables are boolean values.

data		methods
`double mwPred` M_W	pole M_W prediction (GeV)	`setMw` `displayMw`
`double msusy` M_S	Minimisation scale (GeV)	`displayMsusy` `setMsusy`
`QedQcd dataset`	M_Z boundary condition on Standard Model couplings	`setData` `displayDataSet`
`sProblem problem`	problem flags	`displayProblem` `flagIrqfp`, `flagB` `flagNonperturbative` `flagTachyon`, `flagHiggsufb` `flagNoConvergence` `flagNoMuConvergence` `flagNoRhoConvergence` `flagMusqwrongsign` `flagAllProblems`
`DrBarPars forLoops`	$\overline{DR}$ masses and mixings	`displayDrBarPars` `setDrBarPars`
`sPhysical physpars`	pole masses and mixings	`displayPhys` `setPhys`

Table 11: `MssmSoftsusy` class data and accessor methods.

in the `bcPars(n+3) DoubleVector`, with the $(n+1, n+2, n+3)^{th}$ element of `bcPars` being the fine-tuning with respect to the Higgs potential parameters (μ and B) and the top Yukawa coupling (h_t) respectively. `fineTune` is an optional feature. `sinSqThetaEff()` returns a `double` number corresponding to a full one-loop calculation of the quantity $\sin^2\theta^l_{eff}$. It does *not* contain any 2-loop corrections, and may not be accurate enough for precision electroweak fits.

E.8 `MssmSoftsusyAltEwsb` Class

The `MssmSoftsusyAltEwsb` class, which inherits directly from `MssmSoftsusy`, adds two private

name	function
`lowOrg`	Driver routine for whole calculation*
`methodBoundaryCondition`	Boundary condition for derived objects*
`itLowsoft`	Performs the iteration between M_Z and unification scale
`sparticleThresholdCorrections`	$\overline{DR}$ radiative corrections to Standard Model couplings at M_Z
`physical`	Calculates sparticle pole masses and mixings
`calcDrBarPars`	Calculates $\overline{DR}$ pole masses and mixings
`rewsb`	Sets μ, B from EWSB conditions
`fineTune`	Calculates fine-tuning for soft parameters* and h_t
`getVev`	Calculates VEV $v^{\overline{DR}}$ at current scale from Z self-energy and gauge couplings
`calcSinthdrbar`	Calculates $s_W^{\overline{DR}}$ at current scale from gauge couplings
`calcMs`	Calculates M_S
`printShort`	short list of important parameters printed out to standard output in columns
`printLong`	long list of important parameters printed out to standard output in columns
`outputFcncs`	prints a list of flavour-changing δ parameters*
`lesHouchesAccordOutput`	prints output in SLHA [29] format
`sinSqThetaEff`	calculates then returns $\sin^2 \theta^l_{eff}$

Table 12: `MssmSoftsusy` methods and related functions. Functions marked with an asterisk are mentioned in the text.

data variables: `muCond` and `mAcond`, both massive parameters in units of GeV. They fix the boundary conditions on $\mu(M_{SUSY})$ and the pole pseudo-scalar Higgs mass m_A(pole). In this case, $m_A(M_{SUSY})$ is extracted from the input m_A(pole) via $m_A^2(M_{SUSY}) = m_A^2(\text{pole}) + \Pi_{AA}(M_{SUSY})$, where $\Pi_{AA}(M_{SUSY})$ is the MSSM pseudo-scalar self-energy correction. The Higgs mass squared parameters $m^2_{H_{1,2}}$ are *not* set at M_{GUT}: they are instead set at M_{SUSY} by solving the simultaneous equations 3.11, 3.12 and using the relationship between $m_3^2(M_{SUSY})$ and $m_A(M_{SUSY})$:

$$\begin{aligned} m^2_{H_1} &= \sin^2\beta(m_A^2 + M_Z^2) + \frac{t_1}{v_1}(1-\sin^4\beta) - \sin^2\beta\cos^2\beta\frac{t_2}{v_2} - (\mu^2 + \frac{1}{2}M_Z^2) \\ m^2_{H_2} &= \cos^2\beta(m_A^2 + M_Z^2) + \frac{t_2}{v_2}(1-\cos^4\beta) - \sin^2\beta\cos^2\beta\frac{t_1}{v_1} - (\mu^2 + \frac{1}{2}M_Z^2), \end{aligned} \quad \text{(E.4)}$$

where all quantities in eq. E.4 are running parameters evaluated at M_{SUSY}. This option is covered under the SLHA input parameters `EXTPAR 23,26` [29].

data		methods
`flavourPhysical fv`	sfermion mixing/mass data	`setFlavourPhys` `displayFlavourPhys`

`double theta12, theta23`
`double theta13, deltaCkm`
$\theta_{12}, \theta_{23}, \theta_{13}, \delta$
	CKM matrix parameters	`setTheta12`
		`setTheta13`
		`setTheta23`
		`setDelta`
		`displayTheta12`
		`displayTheta23`
		`displayTheta13`
		`displayDelta`

`double thetaB12, thetaB23`
`double thetaB13`
$\bar{\theta}_{12}, \bar{\theta}_{13}, \bar{\theta}_{23}$
	PMNS matrix parameters	`setThetaB12`
		`setThetaB13`
		`setThetaB23`
		`displayThetaB12`
		`displayThetaB23`
		`displayThetaB13`

`double mNuE, mNuMu, mNuTau`
$m_{\nu_e}, m_{\nu_\mu}, m_{\nu_\tau}$
	light neutrino masses	`setMnuE`
		`setMnuNu`
		`setMnuTau`
		`displayMnuE`
		`displayMnuMu`
		`displayMnuTau`

`double mcMc`
$m_c(m_c)$
`double md2GeV, mu2GeV, ms2GeV`
$m_d(2), m_u(2), m_s(2)$
	running charm quark mass
	
	quark masses at 2 GeV
		`setMcMc`
		`displayMcMc`
		`setMd2GeV`, `setMu2GeV`
		`setMs2GeV`
		`displayMd2GeV`
		`displayMu2GeV`
		`displayMs2GeV`

`double mePole, mmuPole`
m_e^{pole}, m_μ^{pole}
	pole lepton masses	`setPoleMe, setPoleMmu`
		`displayPoleMe`
		`displayPoleMmu`

Table 13: `FlavourMssmSoftsusy` class data and accessor methods. All angles are measured in radians and masses are measured in GeV.

E.9 `FlavourMssmSoftsusy` Class

`FlavourMssmSoftsusy` objects inherit directly from `MssmSoftsusy` with the addition of the variables shown in Table 13. Some of these constitute the angles of the PMNS and CKM matrices in the standard parameterisation of eq. 3.7. The `lesHouchesAccordOutput` method has been overloaded to provide flavour violating input and output in accordance with the SLHA2 conventions [40]. Note that `SOFTSUSY` currently does *not* contain CP-violating

complex phases, despite the inclusion of δ. Thus, when CKM angles are input via the SLHA2 in the Wolfenstein parameterisation, the magnitude of the 13 entry is fit to $\sin\theta_{13}$ as in eq. 3.8 and δ is set to zero (for now). The relevant method is

```
void FlavourMssmSoftsusy::setAngles
(double lambda, double aCkm, double rhobar, double etabar)
```

Currently, an identical parameterisation is used for the PMNS matrix that describes lepton mixing, except θ_{ij} is replaced by $\bar{\theta}_{ij}$. There is currently no provision for a CP-violating phase. Effective light neutrino masses are also included in the object.

An optional feature intended for studies of flavour-changing neutral currents (FCNCs) is the method

```
void FlavourMssmSoftsusy::sCkm
(DoubleMatrix & deltaULL, DoubleMatrix & deltaURR, DoubleMatrix & deltaULR,
 DoubleMatrix & deltaDLL, DoubleMatrix & deltaDRR, DoubleMatrix & deltaDLR)
const
```

which calculates the parameters $(\delta^{u,d}_{LL,LR,RR})_{ij}$ calculated in the mass-squared-insertion approximation (after a rotation to the super CKM basis), as defined in Ref. [39].

The relationship between the super CKM basis of the mass matrices (including one-loop corrections) and the pole-mass basis as described by SLHA2, is contained in the structure `flavourPhysical`. We list the relevant data in Table 14.

data	description
`DoubleMatrix dSqMix`	6×6 down squark mixing
`DoubleMatrix uSqMix`	6×6 up squark mixing
`DoubleVector msD`	6 mass-ordered down squark masses
`DoubleVector msU`	6 mass-ordered up squark masses
`DoubleMatrix eSqMix`	6×6 charged slepton mixing matrix
`DoubleMatrix nuSqMix`	3×3 sneutrino mixing matrix
`DoubleVector msE`	6 mass-ordered charged slepton masses
`DoubleVector msNu`	3 mass-ordered sneutrino masses

Table 14: `flavourPhysical` structure. All mass parameters are pole masses, stored in units of GeV. The mixing matrix definitions exactly coincide with those in the SLHA2 [40] and describe the transformation between the loop-corrected super CKM basis and the mass basis.

Acknowledgments

This work has been partially supported by PPARC, STFC and the Aspen Center for Physics. We would like to thank A. Djouadi, S. Kraml and W. Porod for help with detailed comparisons [9] with the codes `SUSPECT`, `ISAJET` and `SPHENO`, P. Athron, G. Bélanger, F. Boudjema, M. Eads, J. Hetherington, M. Ibe, J. Kersten, S. Kom, S. Martin, N. Mahmoudi, A. Pukhov, M. Ramage, R. Ruiz, S. Sekmen, A. Sheplyakov, P. Slavich, P. Skands,

J. Smidt, S. Tetsuo, T. Watari and A. Wingerter for bug finding and coding suggestions, I. Gafton, J. Holt, F. Krauss, D. Sanford and F. Yu for suggestions on the draft, D. Ross for information on the Passarino-Veltman integrals and K. Matchev for other useful discussions.

References

[1] G. G. Ross and R. G. Roberts, *Minimal supersymmetric unification predictions*, *Nucl. Phys.* **B377** (1992) 571–592.

[2] V. Barger, M. S. Berger, and P. Ohmann, *The supersymmetric particle spectrum*, *Phys. Rev.* **D49** (1994) 4908–4930, [hep-ph/9311269].

[3] D. M. Pierce, J. A. Bagger, K. Matchev, and R. jie Zhang, *Precision corrections in the minimal supersymmetric standard model*, *Nucl. Phys.* **B491** (1997) 3–67, [hep-ph/9606211].

[4] **MSSM Working Group** Collaboration, J. K. A. Djouadi and G. Moultaka, *Suspect: a program for the supersymmetric spectrum*, hep-ph/9901246.

[5] H. Baer, F. E. Paige, S. D. Protopopescu, and X. Tata, *Isajet 7.48: A monte carlo event generator for p p, anti-p p, and e+ e- reactions*, hep-ph/0001086.

[6] B. C. Allanach, J. P. J. Hetherington, M. A. Parker, and B. R. Webber, *Naturalness reach of the large hadron collider in minimal supergravity*, *JHEP* **08** (2000) 017, [hep-ph/0005186].

[7] G. Corcella *et. al.*, *Herwig 6: An event generator for hadron emission reactions with interfering gluons (including supersymmetric processes)*, *JHEP* **01** (2001) 010, see the HERWIG web page http://hepwww.rl.ac.uk/theory/seymour/herwig/, [hep-ph/0011363].

[8] B. C. Allanach, *SUSY Predictions and SUSY Tools at the LHC*, arXiv:0805.2088 [hep-ph].

[9] B. C. Allanach, *Theoretical Uncertainties in Sparticle Mass Predictions*, Snowmass Proceedings 2001, [hep-ph/0110227]; B. C. Allanach *et al. The Beyond the Standard Model Working Group: Summary Report*, Les Houches proceedings 2002, [hep-ph/0204031]. B. C. Allanach, S. Kraml and W. Porod, *Comparison of SUSY mass spectrum calculations*, SUSY 2002 proceedings, [hep-ph/0207314]; B. C. Allanach, S. Kraml and W. Porod, *Theoretical uncertainties in sparticle mass predictions from computational tools*, JHEP **0303** (2003) 016, hep-ph/0302102; B. C. Allanach, G. Bélanger, F. Boudjema, A. Pukhov and W. Porod, *Uncertainties in Relic Density Calculations in mSUGRA*, hep-ph/0402161; *ibid* in *Beyond the Standard Model Working Group: Summary Report*, Les Houches "Physics at TeV Colliders 2003", hep-ph/0402295.

[10] F. Gabbiani, E. Gabrielli, A. Masiero, and L. Silvestrini, *A complete analysis of fcnc and cp constraints in general susy extensions of the standard model*, *Nucl. Phys.* **B477** (1996) 321–352, [hep-ph/9604387].

[11] S. G. Gorishnii, A. L. Kataev, S. A. Larin, and L. R. Surguladze, *Corrected three loop qcd correction to the correlator of the quark scalar currents and gamma (tot) (h0 $\rightarrow$ hadrons)*, *Mod. Phys. Lett.* **A5** (1990) 2703–2712.

[12] O. V. Tarasov, A. A. Vladimirov, and A. Y. Zharkov, *The gell-mann-low function of qcd in the three loop approximation*, *Phys. Lett.* **B93** (1980) 429–432.

[13] S. G. Gorishnii, A. L. Kataev, and S. A. Larin, *Next next-to-leading perturbative qcd corrections and light quark masses*, *Phys. Lett.* **B135** (1984) 457.

[14] K. G. Chetyrkin, B. A. Kniehl, and M. Steinhauser, *Strong coupling constant with flavour thresholds at four loops in the ms-bar scheme*, *Phys. Rev. Lett.* **79** (1997) 2184–2187, [hep-ph/9706430].

[15] L. V. Avdeev and M. Y. Kalmykov, *Pole masses of quarks in dimensional reduction*, *Nucl. Phys.* **B502** (1997) 419, [hep-ph/9701308].

[16] A. Bednyakov, A. Onishchenko, V. Velizhanin and O. Veretin, *Two-loop $O(\alpha_s^2)$ MSSM corrections to the pole masses of heavy quarks*, [hep-ph/0210258];

[17] **Particle Data Group** Collaboration, see the latest Review of Particle Physics on [http://pdg.lbl.gov/]

D. E. Groom *et. al.*, *Review of particle physics*, *Eur. Phys. J.* **C15** (2000) 1.

[18] J. A. Casas, J. R. Espinosa, and H. E. Haber, *The higgs mass in the mssm infrared fixed point scenario*, *Nucl. Phys.* **B526** (1998) 3–20, [hep-ph/9801365].

[19] S. P. Martin, *Two-loop effective potential for a general renormalizable theory and softly broken supersymmetry*, *Phys. Rev.* **D65** (2002) 116003 [hep-ph/0111209].

[20] Y. Yamada, *Two-loop renormalization of tan(beta) and its gauge dependence*, *Phys. Lett.* **B530** (2002) 174 [hep-ph/0112251].

[21] A. Dedes and P. Slavich, *Two-loop corrections to radiative electroweak symmetry breaking in the mssm*, hep-ph/0212132.

[22] A. Katsikatsou, A. B. Lahanas, D. V. Nanopoulos, and V. C. Spanos, *On the radiative corrections to the pseudo-scalar higgs boson mass*, *Phys. Lett.* **B501** (2001) 69–77, [hep-ph/0011370].

[23] G. Degrassi, P. Slavich, and F. Zwirner, *On the neutral higgs boson masses in the mssm for arbitrary stop mixing*, *Nucl. Phys.* **B611** (2001) 403–422, [hep-ph/0105096].

[24] A. Brignole, G. Degrassi, P. Slavich, and F. Zwirner, *On the o(alpha**2(t)) two-loop corrections to the neutral higgs boson masses in the mssm*, *Nucl. Phys.* **B631** (2002) 195–218, [hep-ph/0112177].

[25] A. Brignole, G. Degrassi, P. Slavich, and F. Zwirner, *On the two-loop sbottom corrections to the neutral higgs boson masses in the mssm*, *Nucl. Phys.* **B643** (2002) 79–92, [hep-ph/0206101].

[26] A. Dedes, G. Degrassi and P. Slavich, *On the two-loop Yukawa corrections to the MSSM Higgs boson masses at large tan(beta)*, hep-ph/0305127.

[27] R. Barbieri and A. Strumia, *About the fine-tuning price of lep*, *Phys. Lett.* **B433** (1998) 63–66,

[28] B. de Carlos and J. A. Casas, *One loop analysis of the electroweak breaking in supersymmetric models and the fine tuning problem*, *Phys. Lett.* **B309** (1993) 320–328, [hep-ph/9303291].

[29] P. Skands *et al*, *Interfacing SUSY Spectrum Calculators, Decay Packages and Event Generators*, *JHEP* **0407** (2004) 036, [hep-ph/0311123].

[30] B. C. Allanach, A. Djouadi, J. L. Kneur, W. Porod and P. Slavich, *Precise determination of the neutral Higgs boson masses in the MSSM* hep-ph/0406166; B. C. Allanach, G. A. Blair, S. Kraml, H. U. Martyn, G. Polesello, W. Porod and P. M. Zerwas, *Reconstructing supersymmetric theories by coherent LHC / LC analyses*, hep-ph/0403133.

[31] S. P. Martin and M. T. Vaughn, *Two Loop Renormalization Group Equations For Soft Supersymmetry Breaking Couplings*, Phys. Rev. D **50** (1994) 2282 [arXiv:hep-ph/9311340].

[32] H. Baer, J. Ferrandis, K. Melnikov and X. Tata, *Relating bottom quark mass in $\overline{DR}$ and $\overline{MS}$ schemes*, *Phys. Rev.* **D66** (2002) 074007, [hep-ph/0207126].

[33] B.C. Allanach *et al*, *The Snowmass Points and Slopes: Benchmarks for SUSY Searches*, *Eur. Phys. J.* **C25** (2002) 113, [hep-ph/0202233].

[34] M. Muhlleitner, A. Djouadi and Y. Mambrini, *SDECAY: A Fortran code for the decays of the supersymmetric particles in the MSSM* [hep-ph/0311167].

[35] T. Sjostrand, L. Lonnblad, S. Mrenna and P. Skands, *PYTHIA 6.3 physics and manual*, [hep-ph/0308153].

[36] G. Belanger, F. Boudjema, A. Pukhov and A. Semikov, *micrOMEGAs: A program for calculating the relic density in the MSSM*, *Comput. Phys. Comm.* **149** (2002) 103, [hep-ph/0112278].

[37] L. Randall and R. Sundrum, *Out of this world supersymmetry breaking*, *Nucl. Phys.* **B557** (1999) 79, [hep-th/9810155].

[38] S. Ambrosanio, G. D. Kribs and S. P. Martin, *Signals for gauge-mediated supersymmetry breaking models at the CERN LEP2 collider*, *Phys. Rev.* **D56** (1997) 1761 [hep-ph/9703211].

[39] F. Gabbiana, E. Gabrielli, A. Masiero and L. Silvestrini, *A Complete Analysis of FCNC and CP Constraints in General SUSY Extensions of the Standard Model*, *Nucl. Phys.* **B477** (1996) 321 [hep-ph/9604387].

[40] B.C. Allanach *et al*, *SUSY Les Houches Accord 2*, arXiv:0801.0045 [hep-ph].